

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

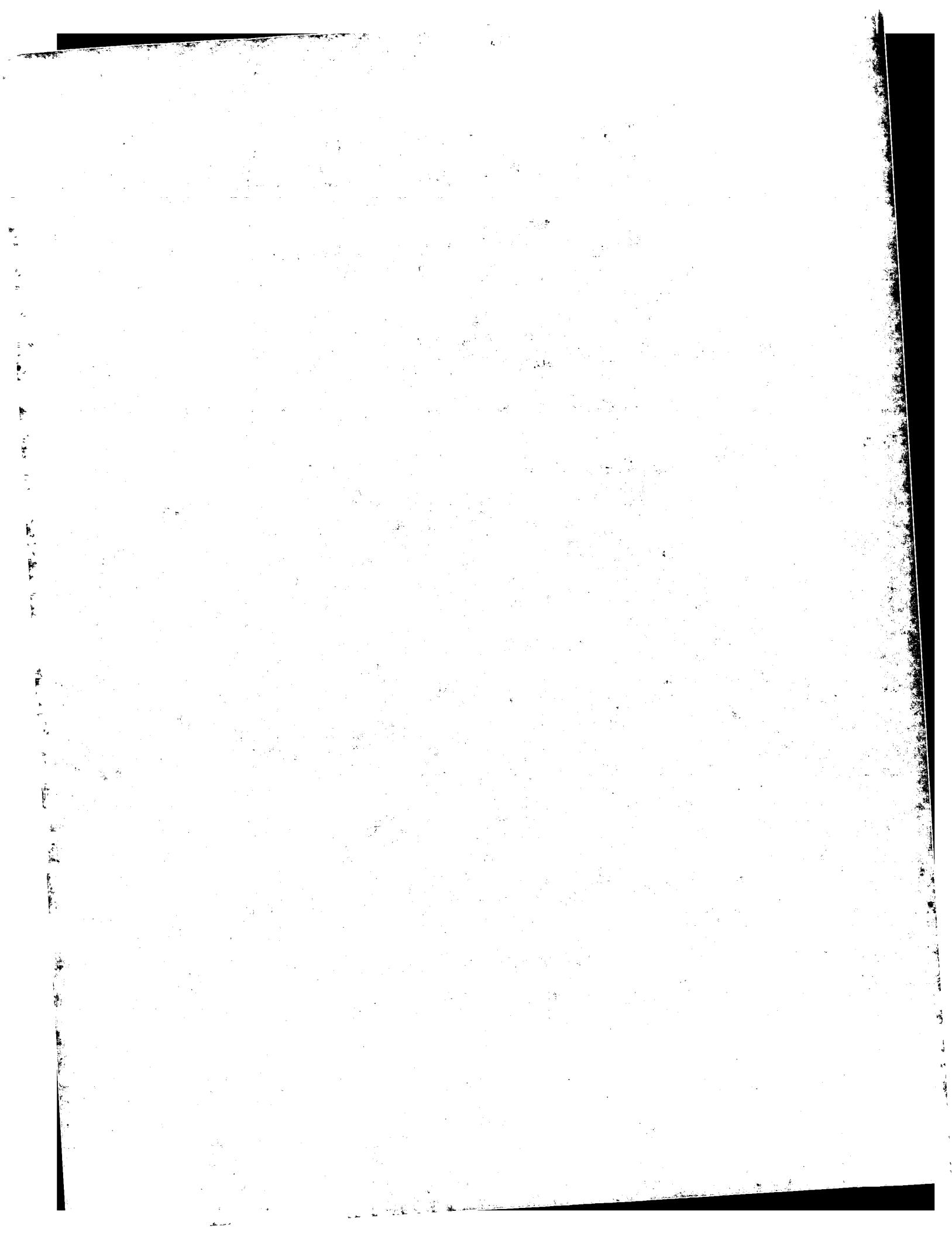
Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

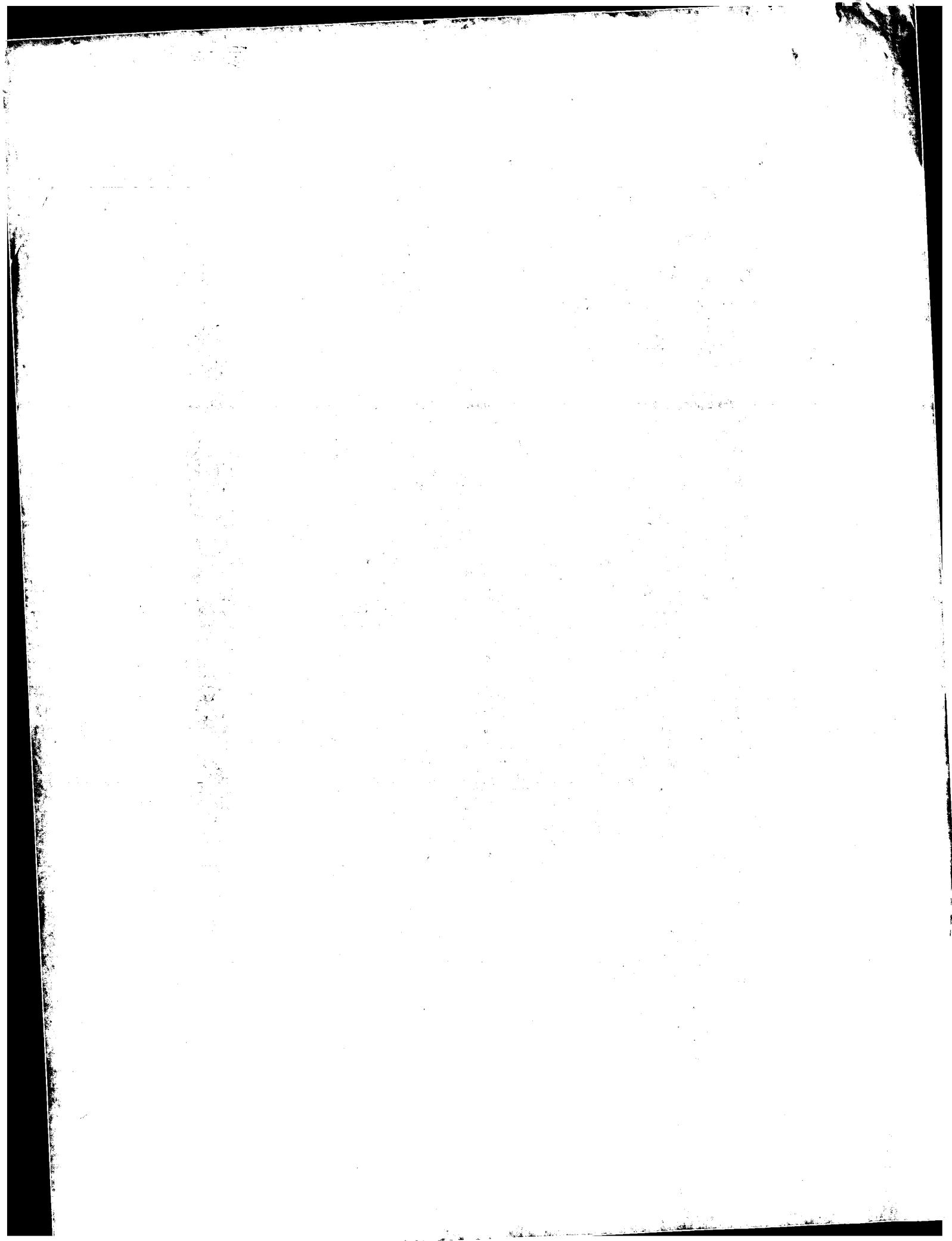
**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Problem Image Mailbox.**



**Japanese Reference X**  
(Statement of Relevance)

This reference was cited during an opposition proceeding for a Japanese patent (JP 2995156 granted 11/22/99) issuing from an application corresponding to U.S. patent application Serial No. 07/475,111.



# パソコンで学ぶ パターン認識と图形処理

附录

上坂吉則	
1959年	名古屋工業大学電気工学科卒業
1961年	NHK NHK音楽専門研究会にて
1971年	NHK NHK音楽専門研究会にて
1973年	日本大学音楽専門（准教授）
1979年	東京藝術大学音楽（准教授）
現在に至る。	工学博士
主著書	
「ハーモニック・オーディションの実践」（筑波社、1977）	
「情報音楽の基礎」（筑波社、1977）	
「情報音楽の実践」（筑波社、1979）	
太郎背夫	
1974年	筑波大学工芸部電子工学科卒業
1979年	筑波大学工芸部電子工学科修士課程修了
現在に至る。	工芸博士
筑波大学助手（工芸学部）	
筑波大学助手（工芸学部）	
筑波大学助手（工芸学部）	
筑波大学助手（工芸学部）	

著者　太原吉坂

バターン認識と图形処理

卷之三

印  
鑄  
省

1984年11月30日 第1刷発行  
1985年12月20日 第2刷発行

免行所

中162 賴文正著《五胡內亂》34~8  
中163 賴文正著《五胡內亂》34~3

農村印記：日本

© 1984  
Printed in JAPAN

# 11 パターン・マッチング——DPマッチング

テンプレートの変形をある出力許容しながら入力パターンとのマッチングを行なう方法を考える。ミスマッチの割度を計算する過程での計画法が用いられる。

## 明 説

パターン・マッチング（場合）は、あらかじめ用意されたテンプレート（参照パターン）を入力パターンに重ね合わせ、それらが一致するかどうかによって入力パターンの識別を行なう最も基本的なパターン認識の手法である。しかし、原理は簡単であっても、マッチングの方法やマッチングの度合の評価にはいろいろなもののが考えられる。

では、入力パターンに対し、テンプレートをそのまま重ね合わせてマッチングを行なう最も簡単なマッチング法について述べた。この章では、テンプレートをある制限のもとで変形せながら入力パターンと重ね合わせ、最もよくマッチしたときのずれの量（パターン間の距離）を計算する。

テンプレートを  $F$ 、入力パターンを  $G$  として  $N$  次元の時系列パターンを考えることにする：

$$F: F(t) = (F_1(t), F_2(t), \dots, F_N(t)) \quad (0 \leq t \leq T_F)$$

$$G: G(t) = (G_1(t), G_2(t), \dots, G_N(t)) \quad (0 \leq t \leq T_G)$$

時系列パターンの変形として最も一般的なものは、時間軸上の変動による変形、すなわち伸縮変形である。たとえば、音声パターンは時系列パターンの代表的なものであるが、人間が音楽を話す速度は一定ではないので、同じ人が同じ単語を発声してもその速度得られる音声パターンには時間軸上の伸縮変形が生じることになる。このようなパターンに対してマッチングを行なう場合には、時間軸上の伸縮変形を補正し、テンプレートと入力パターンが最も近似した状態になるようにしてから両者の間のずれを計算すべきであると考えられる。 $u(t)$  は図11.1に示すよ

うな連続な单調増加関数で

$$u(0) = 0.$$

$$u(T_0) = T_F$$

である。このように時間軸を変換したテンプレート  $F(u(t))$  と入力パターン  $G(t)$  の間の距離  $E(F, G, u(t))$  を

$$E(F, G, u(t)) = \frac{\int |G(t) - F(u(t))| ds}{\int ds} \quad (11.1)$$

と定義することにしよう。ここで

$$\|G(t) - F(u(t))\| = \sqrt{\sum_{n=1}^N (G_n(t) - F_n(u(t)))^2},$$

$$ds = \sqrt{1 + \left(\frac{du}{dt}\right)^2} dt$$

であり、積分は  $0 \leq t \leq T_0$  について行なうものとする。この  $E(F, G, u(t))$  を最小にするよう  $u(t)$  を選択すれば、 $F$  と  $G$  が最も近似した状態になるはずである。したがって、そのときの  $E(F, G, u(t))$  の値を  $F$  と  $G$  の距離  $D(F, G)$  とすればよい。

$$D(F, G) = \min_{u(t)} [E(F, G, u(t))] \quad (11.2)$$

さて、テンプレート  $F$ 、入力パターン  $G$  がともに時間軸上で標準化された時系列パターンである場合について考えてみよう：

$$F = f_1, f_2, \dots, f_p, \dots, f_n,$$

$$G = g_1, g_2, \dots, g_p, \dots, g_n.$$

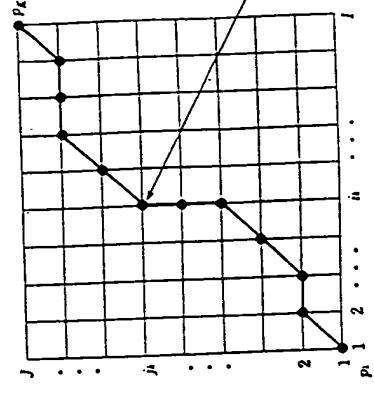


図 11.2 構子グラフ

この場合、時間軸の伸縮変換  $u(t)$  は、図11.2に示すような格子グラフ上の点  $(j, j)$  から点  $(l, j)$  へ至るルートによって表現される。したがって、マッチングにおける  $F$  の標本点と  $G$  の標本点の対応関係  $P$  はこのルート上の点列で与えられることになる。格子点を  $p_i = (i, j_s)$   $(1 \leq i, l \leq l, 1 \leq j_s \leq j)$

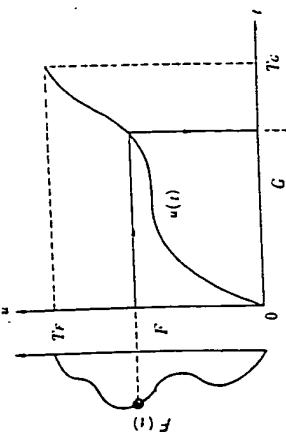


図 11.1 関数  $u(t)$  による時間軸の伸縮変換

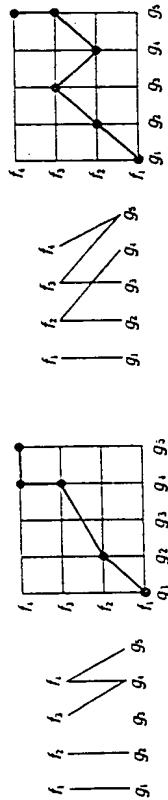
テンプレートと入力パターンが最も近似した状態になるようにしてから両者の間のずれを計算することにする。 $u(t)$  は図11.1に示すよ

とすれば

$$P = p_0, p_1, \dots, p_K$$

である。時間軸の伸縮変換が不自然なものにならないようにするため、ルートの選択には少なくともつぎのような制限が加えられる：

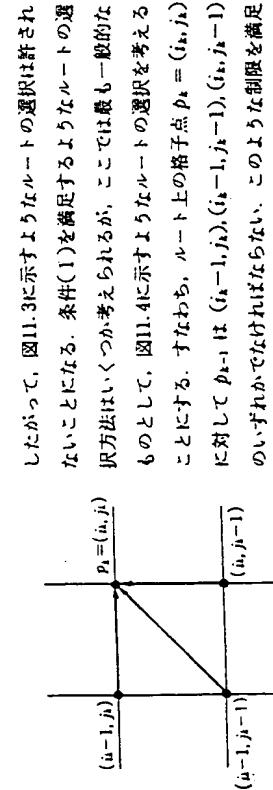
(1) 連続性と単調増加性。

(2)  $p_1 = (1, 1)$ かつ  $p_K = (I, J)$ 。

(a) 連続性に反する例

(b) 単調増加性に反する例

図 11.3 連続性と単調増加性に反する例

図 11.4 終点  $p_k$  に至るルート

$$D(F, G) = \min_P \left[ \frac{\sum_{k=1}^K d(p_k) d_k}{\sum_{k=1}^K d_k} \right] \quad (11.3)$$

で与えられる。ここで

$$d(p_k) = \|p_k - f_k\|$$

であり、 $d_k$  は (11.1) の  $d_k$  に對応している。具体的に与えられた  $F$  と  $G$  について  $D(F, G)$  の値を求める場合、(11.3) をまとめてそのままで計算したのでは、最小値を得るのに相当な量の計算が必要となる。一般に、このような最小化問題は、動的計画法によって少ない計算量で効率よく解くことができる。 $d_k$  を

$$d_k = (i_k - i_{k-1}) + (j_k - j_{k-1})$$

ただし

$$i_0 = j_0 = 0$$

と定義すれば

$$p_{k-1} = (i_{k-1}, j_{k-1}) \text{ ならば } P(i_k, j_k) = 1,$$

$$\sum_{k=1}^K d_k = I + J$$

であるから、(11.3) は

$$D(F, G) = \frac{1}{I + J} \min_P \left[ \sum_{k=1}^K d(p_k) d_k \right] \quad (11.4)$$

となり動的計画法が適用できる。

標本点間の対応関係  $P$  に関して、つきのような部分和を考える：

$$h(p_k) = \min_{p_{k-1}, \dots, p_1} [d(p_k) d_k + d(p_{k-1}) d_{k-1} + \dots + d(p_1) d_1] \quad (k \leq 2), \quad (11.5)$$

この部分和 (11.5) はただちに漸化式の形に変形でき

$$h(p_k) = \min_{p_{k-1}} [h(p_{k-1}) + d(p_k) d_k] \quad (11.6)$$

となる。ここで  $h(p_1) = d(p_1) d_1 = 2d(p_1)$  である。ところで、格子グラフ上でルートの選択には図 11.4 のような制限があるので、(11.6) は

$$h(p_k) = \min \left\{ \begin{array}{l} h(i_{k-1}, j_{k-1}) + 2d(p_k) \\ h(i_{k-1}, j_{k-1}) + 2d(i_k, j_k) \\ h(i_k, j_{k-1}) + d(i_k, j_k) \end{array} \right\} \quad (11.7)$$

となる。この漸化式により、 $h(p_K)$  は  $h(p_1)$  を初期条件として順次求めることになる。このように、 $h(p_K)$  が求めれば、ミスマッチの測度  $D(F, G)$  は

$$D(F, G) = \frac{1}{I + J} h(p_K) \quad (11.8)$$

となる。

以上のような方法によるバターン・マッチング(11.3)、ミスマッチの測度を計算する過程で動的計画法 (Dynamic Programming) が用いられるので、DP マッチングと一般に呼ばれている。

### 11.3 プログラム

テンプレート  $F$ 、入力バターン  $G$  として 1 次元の時系列パターンが与えられたとき、DP プログラムにより、 $F$  と  $G$  のミスマッチの測度  $D(F, G)$  を求めるプログラムを考える。

一般に、DP マッチングを用いて入力バターンの認識を行なう場合には、単に  $D(F, G)$  の値が求まりさえすればよく、そのときの対応関係  $P$  は知る必要がないので、漸化式 (11.7) をそのまま計算するだけでよい。しかし、本章の目的は DP マッチングそのものを理解することにあるので、このプログラムでは、 $D(F, G)$  を与える対応関係  $P$  がどのようにになっているのかということも表示することにする。対応関係  $P$  を求めるには、配列  $P(i_k, j_k)$  ( $1 \leq i_k \leq I, 1 \leq j_k \leq J$ ) を用意しておき、漸化式 (11.7) の計算において最小値を与える格子点  $p_{k-1}$  の  $p_k$  に対する相対位置をその部度記憶しておけばよい。すなはち、 $h(p_k)$  を求めたとき最小値を与える  $p_{k-1}$  が、

$$\begin{aligned} p_{k-1} &= (i_{k-1}, j_k) \text{ なら } P(i_k, j_k) = 2, \\ p_{k-1} &= (i_k, j_{k-1}) \text{ なら } P(i_k, j_k) = 3 \end{aligned}$$

としておけばよい。DPマッチングの終了後,  $i_k = i_{k-1}, j_k = j_{k-1}$  を初期条件として, 順次

$$\begin{aligned} P(i_k, j_k) &= 1 \quad \text{なら } i_{k-1} = i_k - 1, \quad j_{k-1} = j_k - 1 \\ P(i_k, j_k) &= 2 \quad \text{なら } i_{k-1} = i_k - 1, \quad j_{k-1} = j_k \\ P(i_k, j_k) &= 3 \quad \text{なら } i_{k-1} = i_k, \quad j_{k-1} = j_k - 1 \end{aligned} \quad (11.9)$$

することにより, 格子点  $p_k = (i_k, j_k)$  ( $1 \leq k \leq K$ ) を求めれば,  $D(F, G)$  を与える対応関係  $P$  が得られる。

プログラムの全体の流れはつきのようになっている:

1° 製題の表示。

2° 初期設定。

3° テンプレートと入力バターンの指定。

指定されたデータ番号により, プログラム末尾にある時系列バターンのデータのなかからテンプレート  $F$  とするバターン, 入力バターン  $G$  とするバターンを読み込む。読み込んだデータはただちにディスプレイ上に表示する。

4° DPマッチングの計算。

既化式 (11.7) を用いてミスマッチの測度  $D(F, G)$  を計算する。また,  $D(F, G)$  を与える対応関係  $P$  を求めるため, この計算と平行して  $h(p_k)$  を与える格子点  $p_{k-1}$  の相対位置を配列  $P(i_k, j_k)$  に記憶しておく。

5° 対応関係の表示。

ミスマッチの測度  $D(F, G)$  を与える対応関係  $P$  を (11.9) に従って求め, 格子グラフ上のルートとして表示する。

6° 結果の表示。

テンプレート  $F$ , 入力バターン  $G$  の時系列バターンの標本点が具体的にどのように対応付けられたのか表示し, かつ計算の結果得られたミスマッチの測度  $D(F, G)$  の値を表示する。

なお, このプログラムではグラフィック表示を用いるので, 実行の際には付録Aのグラフィック・サブルーチンをアヘンドしていただきたい。

プログラムで用いている主な変数はつきのとおりである:

$F$  = 大きさ  $NF$  の1次元配列でテンプレート  $F$  用の時系列バターン。

$NF$  = テンプレート  $F$  の標本数。

$G$  = 大きさ  $NG$  の1次元配列で入力バターン  $G$  用の時系列バターン。

$NG$  = 入力バターン  $G$  の標本数。

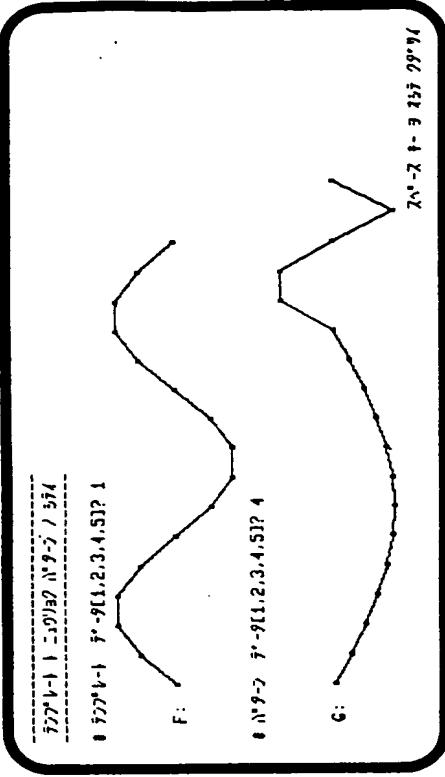
$H$  = (1, 5) で定義される  $h(i_k, j_k)$  を表す。

$P$  = 対応関係  $P$  を求めるための配列  $P(i_k, j_k)$  を表す。

X7, X8, X9, Y7, Y8, Y9, X7%6, X8%, Y7%, Y8% = グラフィック・サブルーチン用の

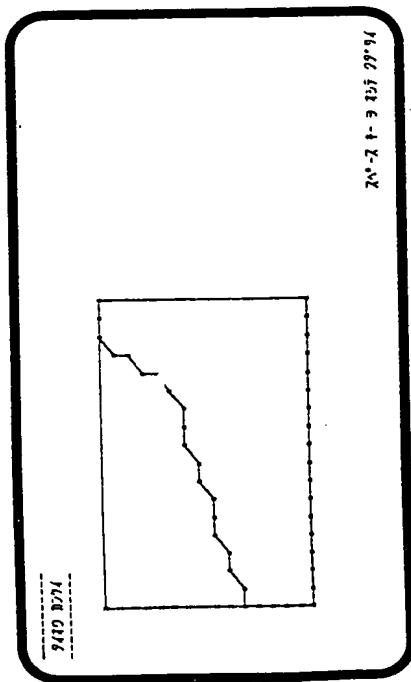
入力変数 (付録A参照)。

実行結果



プログラムの実行例を以下に示す。プログラムを実行すると, まず製題が表示される。スペース・キーを押すと, プログラム末尾にある5つの時系列データのうちどれをテンプレート  $F$  とし, どれを入力バターン  $G$  とするか指定するよう要求てくる。ここでは, 例としてデータ番号1の時系列データをテンプレートとし, データ番号4の時系列データを入力バターンとすることにする。データ番号を入力すると, それぞれのデータが読み込まれ表示される。

プログラムの実行例を以下に示す。プログラムを実行すると, まず製題が表示される。スペース・キーを押すと, プログラム末尾にある5つの時系列データのうちどれをテンプレート  $F$  とし, どれを入力バターン  $G$  とするか指定するよう要求てくる。ここでは, 例としてデータ番号1の時系列データをテンプレートとし, データ番号4の時系列データを入力バターンとすることにする。データ番号を入力すると, それぞれのデータが読み込まれ表示される。



スペース・キーを押すと、 $F$  と  $G$  の原点が具体的にどのように対応付けられているのか表示される。そして、そのような対応関係における  $D(F, G)$  の値が表示される。 $D(F, G)$  はミスマッチの割度であるから、 $F$  と  $G$  が同一のパターンであればもちろん 0 となり、異なれば異なるほど大きな値となる。プログラムの末尾にある 5 つの時系列・パターンのデータについて比較検討してみるとよい。

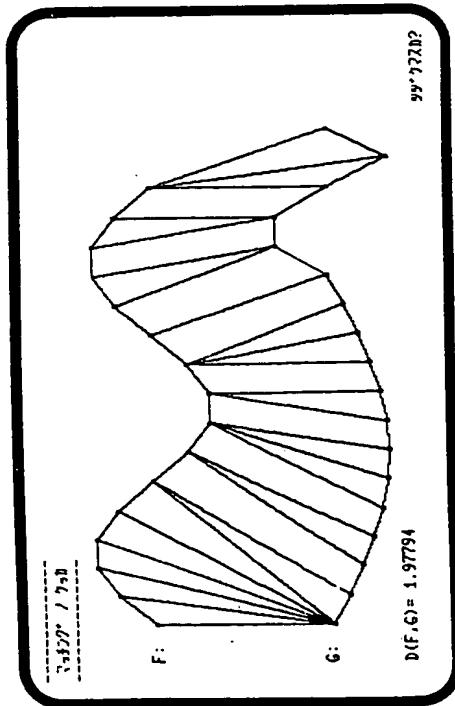


図 11.5 スプリング負荷モデルを用いたパターン・マッチング

1 つの方法は、2 次元平面上の幾何学パターンをフリーマンのチャーン・コード (11 参照) や全曲率関数 (18 参照) などを用いて 1 次元の配列あるいは 1 变数の関数で表現する方法である。このような表現をすれば、2 次元平面上のパターンも時系列・パターンと同様な取り扱いが可能になり、DP マッチングをそのまま適用することができる。

もう 1 つの方法は、テンプレートの変形のしかたを 2 次元に拡張する方法である。この場合、一般にテンプレートとしてはスプリング負荷モデルが用いられる。スプリング負荷モデルは、図 11.5 に示すように、いくつかの部分・パターンをスプリングで結合した構造をしている。このようなテンプレートを用いてマッチングを行なう場合には、スプリング負荷モデルのスプリングを伸縮しながら入力パターンに重ね合わせ、最もよくマッチしたとき、すなわち各部分・パターン自体がそれぞれよくマッチし、かつスプリングにかかる負荷が全体としてできるだけ少ないときのずれの量を求め、それをミスマッチの割度とすればよい。

いま、テンプレート  $F$  として  $N$  個の部分・パターン ( $\#1 \sim \#N$ ) からなるスプリング負荷モデルを考え、入力パターン  $G$  が定義されている 2 次元平面上に重ね合わせるものとする (図 11.5 参照)。部分・パターン  $\#i$  の位置を  $r_i = (x_i, y_i)$  としたとき、部分・パターン・コードの強度を  $d_i(r_i)$  とする。また部分・パターン  $\#i$  が  $r_i = (x_i, y_i)$  にあり、部分・パターン  $\#j$  が  $r_j = (x_j, y_j)$  にあつたとき、両者を結合しているスプリングにかかる負荷を  $g_{ij}(r_i, r_j)$  とする。この  $g_{ij}(r_i, r_j)$  としては、たとえば  $k_{ij}, l_{ij}$  を定数として

$$g_{ij}(r_i, r_j) = k_{ij}(|r_i - r_j| - l_{ij})$$

というような関数を考えればよい。さて、 $N$  個の部分・パターンの位置を  $R_N$  としたとき、部分・パターン  $\#i$  から  $\#j$  の距離を  $R_{ij}$  で表現することにする：

$$R_{ij} = [r_i, r_j, \dots, r_i]$$

このとき、位置  $R_i$  に関する部分・パターン  $\#i$  のずれの量  $h_i(R_i)$  を

